Service-Oriented Grid Computation for Large-Scale Parameter Estimation in Complex Environmental Modeling

Kejing He*
Guangdong Key Laboratory of Computer Network
South China University of Technology
Guangzhou, 510641, China
kjhe@scut.edu.cn

Shoubin Dong
Guangdong Key Laboratory of Computer Network
South China University of Technology
Guangzhou, 510641, China
sbdong@scut.edu.cn

Li Zheng
Center for Agriculture Resources Research
IGDB, Chinese Academy of Sciences
Shijiazhuang, 050021, China
zhengli.lizheng@gmail.com

ABSTRACT

Complex environmental modeling often involves a large number of unknown physical and ecological parameters. Parameter estimation is one of the most difficult steps in many modeling activities. In this paper we present a service-oriented framework, named GGPE-G (Grid-enabled Global optimization for General Parameter Estimation), for efficient parameter estimation in heterogeneous, distributed systems. Being presented as services, the optimization algorithms, the physical and ecological process models and clients can interact with each other by XML message interactions. The proposed approach supports a generic parameter estimation procedure and can be easily applied to different modeling environment. In this paper, we explain the design, architecture, and implementation of GGPE-G in details. We also apply GGPE-G to a complex soil-water-atmosphere-plant modeling system to demonstrate its utility and efficiency.

Keywords

Parameter Estimation, Parallel, Grid, Service-Oriented Architecture (SOA), Modeling

1. INTRODUCTION

Parameter estimation (PE) aims at determining values of unknown model parameters that provide the best fits, in terms of minimum errors or maximum likelihood, between predictions and observations. In general, this requires the solution of a nonlinear and frequently nonconvex optimization problem which is often a computationally intensive and time consuming process. Traditionally, this work is done by trial-and-error method with tremendous manual labor. In the past few decades, a number of dedicated systems have been produced for automatic parameter estimation, such as PEST [5] and UCODE [11].

Solving a PE problem usually requires the integration of various modules into an often heterogeneous and distributed optimization environment. A common requirement is that the modeling and analysis tools should provide programmatic interfaces for the optimization modules. Usually, such modeling and analysis tools are developed by using different technologies and therefore offer different software interfaces. For example, MODFLOW [7], a widely used three-dimensional finite-difference groundwater flow model, doesn't reserve any interface for further development. It means that integration can only be achieved by system calls and specially developed parsers. MSC.Marc, a notable finite element analysis (FEA) program, offers a number of Python interfaces.

A variety of technologies exist for supporting the interactions between incompatible software components. The most generic and commonly used method is integration via system calls and communication via data files. This approach usually relies on the formats of input and output files, and will not work if the simulation modules run in a distributed mode on separate machines over a network or Internet. The lack of standards for data exchange between different file formats means that modelers have to develop extra processing layers almost every time when a new component is introduced. Both PEST and UCODE belong to this category.

Another approach for integration is based on the standard Remote Procedure Call (RPC) protocol, such as CORBA, Java RMI and COM. However, these vendor-specific techniques often result in islands of interoperability instead of the universal interoperability we need. Further more, these technologies are not firewall-friendly, and are difficult to apply in a widely distributed environment such as the Grid. Both of the above-mentioned approaches are tightly-coupled systems that are less flexible and scalable. For example, most PE searches use a local gradient method that has a tendency of converging to a local instead of the global optimum. If in such systems the optimization logic is closely coupled with the process modeling and input/output processing, it will be very difficult if not impossible for the user to replace the local gradient method with a more efficient optimization algorithm.

The emergence of Grid technologies has provided a new

*Please address correspondence to Kejing He at the above email address.

SAC'06 April 23-27, 2006, Dijon, France

solution for integrating PE function with process modeling. The primary target of Grid technologies is to make large-scale, dynamic collaboration of heterogeneous and distributed resources possible. Adopting service-oriented architecture (SOA), by which all resources on the Grid are regarded as web services, different software components can inter-operate easily. There have been many Grid-enabled applications that have been coupled with web services technology. The Geodise project [1, 12] provides a suite of Grid-enabled design optimization, search tools, and CFD analysis packages within the Matlab environment. The Spitfire project [4] within the European Data Grid Project provides a set of secure Grid enabled middleware services for access to a wide range of relational databases. The Resource Aware Visualization Environment (RAVE) project [3] is developing a distributed, collaborative Grid enabled visualization environment that supports automated resource discovery across heterogeneous machines. RAVE runs as a background process using web services, enabling users to share resources with other users rather than commandeering an entire machine.

The purpose of this paper is to present a generic and extensible service-oriented framework for large scale parameter estimation, named GGPE-G (Grid-enabled Global optimization for General Parameter Estimation). By adopting service-oriented architecture, several benefits achieved. First, the collaboration between different optimization modules and different modeling tools becomes much easier. The optimization codes, regardless in what programming language they are written or on what platforms they run, are encapsulated into standard web services and made universally accessible. The tightly-coupled invocations between the optimization modules and the modeling codes are replaced with loosely-coupled, SOAP-based message interactions. Second, deploying GGPE-G to large physical system becomes more convenient. Web services are deployed over standard Internet technologies. This enables Web services to be deployed even over different clusters under different firewalls. What's more, due to the use of existing standards, underlying security (such as SSL) is already built-in.

In Section 2, we describes the architecture of GGPE-G. Section 3 details the implementation of various services. Section 4 demonstrates the utility and efficiency of GGPE-G with an application in modeling of water cycle in a soil-water-atmosphere-plant system. The application involves the estimation of 150 unknown physical parameters. Section 5 draws a conclusion.

2. ARCHITECTUE

2.1 The High-Level Architecture

The GGPE-G is a service-oriented system that helps modelers to cope with large-scale parameter estimation tasks in complex environmental modeling. The GGPE-G architecture, as illustrated in Fig. 1, is based on SOA with standardized interfaces to both clients and a number of services. A set of GGPE-G services facilitate the entire parameter estimation progress. They are: repository service, computation service, monitoring service and optimization service.

Repository Service provides service-oriented interfaces to access and manage the data storage. There are four exposed interfaces, namely query, store, update and delete. The implementation details are described in section 3.1.

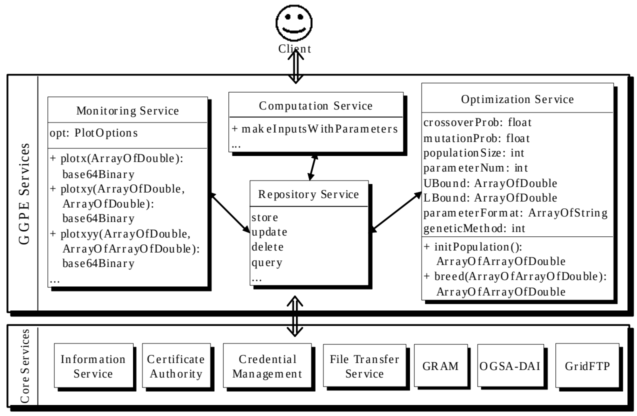

Figure 1: The high-level architecture

Computation Service offers the capability of requesting a set of model runs in a "disconnected" mode, by which the client requests the running of the physical models and then turns over the responsibility of managing model running to the computation service. The client can monitor and change the state of the running through the resource property of computation service. Section 3.2 gives the implementation details of computation service.

Monitoring Service monitors the parameter estimation progress. It interacts with repository service and presents typical variable variances to the user for monitoring the proper progress of the parameter estimation. Implementation details of monitoring service are described in section 3.3.

Optimization Service is the core of GGPE-G. It controls the entire search progress for optimal estimates. Implementation details of optimization service are provided in section 3.4.

We accomplish the above services by employing the core services available from globus toolkit version 4.0 [2], including file transfer service, information services, certificate authority, and credential management etc..

2.2 Security Issues

Grid Security Infrastructure (GSI) provides GGPE-G with both message-level and transport-level security. Message-level security implements the WS-Security standard and the WS-SecureConversation specification to provide message protection for SOAP messages. Features include authentication of the sender, encryption of the message, integrity protection of the message and replay protection. Transport-level security provides a secure channel by using HTTP over SSL/TLS (HTTPS) for transporting the messages. This security mechanism supports all of the security features provided by SSL/TLS with the addition of support for X.509 proxy certificates.

Adopting SOA provides GGPE-G with the greatest capacity to cope with components inter-operation. GGPE-G

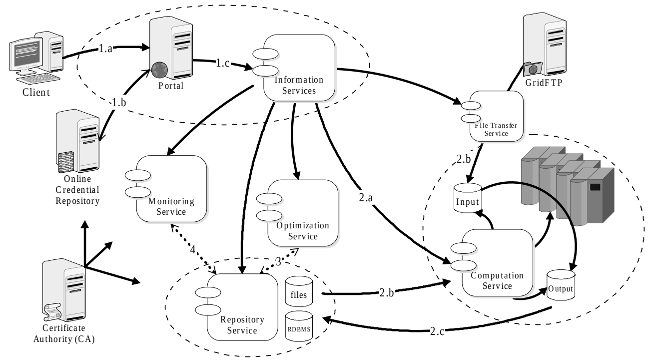

Figure 2: Interactions between clients, services and resources

will, in principle, facilitates the solving of any parameter estimation problems in a service-oriented Grid.

2.3 The Interactions Between Clients, Services and Resources

Clients, Services and Resources interact with each other extensively during the execution of a parameter estimation job. As illustrated in Figure 2, a typical interaction sequence is:

Step 1. (a) User connects to the portal using a standard web browser from any machine and provides username and pass phrase to the portal. (b) Portal contacts credential manager and get short-term proxy credentials using user's username and pass phrase. (c) Portal uses the short-term proxy credential to access Grid on user's behalf.

Step 2. (a) Client invokes a number of computation services asynchronously (i.e., the client invokes the services but does not have to wait for the responses). (b) Computation service obtains required data from repository service and the necessary files from GridFTP respectively. (c) When computation service fulfills its tasks, the parameter estimates and their corresponding errors will be returned to and saved in repository service for further use.

Step 3. Client creates an instance of optimization service. For the specific genetic algorithm we implemented for current study (see section 3.4 for more information), the optimization service will interact with repository service periodically to see whether breeding next generation is needed.

Step 4. Client creates an instance of monitoring service. Monitoring service will interact with repository service periodically and create images from the critical data. The images will be stored in repository through repository service.

Step 5. Through the query interface of repository service, client can obtain the system state, system state images, optimal parameter values and the parameter estimation history.

3. IMPLEMENTATION

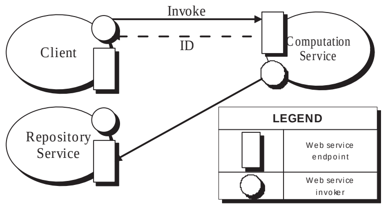

Figure 3: Asynchronous web services

3.1 Repository Service

Repository service provides a way to store and share files and data in a Grid-enabled repository. Each file or datum has some standard metadata associated with it (e.g. the file size and the description etc.).

Structured data, such as system variables and the configuration of the whole parameter estimation job, are stored in native XML format for flexibility. Each file is archived to a GridFTP server. Upon archiving the file is assigned a Universally Unique Identifier (UUID), which allows the file to be physically located and retrieved later. The files' metadata are stored in a relational database for highly efficient data access.

3.2 Computation Service

Since the computation service must perform a series of time-consuming model running, the client cannot afford to pause and wait for the model running to complete before proceeding to the next step. The client can continue with some other tasks rather than wait for the results of model running. The client can also periodically checks the status of the requested jobs using the job IDs generated during the job submission. Fig. 3 illustrates how such an asynchronous working mode is implemented in the computation service.

3.3 Monitoring Service

Parameter estimation of highly nonlinear models is typically time consuming. It is necessary to monitor the optimization progress as it is running. Monitoring may reveal unforeseen problems during the search process, and avoid misleading results and a waste of time. For the optimization by genetic algorithm used in current study, we display a number of standard plots during execution for progress monitoring, including the average fitness plot, the best fitness plot and the convergence. The monitoring service can retrieve data from repository service, post-process the data and then return the images of these plots to repository service. Through repository service, the monitoring images can be accessed and retrieved independently and asynchronously at any platform from any location.

3.4 Optimization Service

There are many methods available for optimization. The most well-known methods fall into three classes: direct search methods, gradient based methods and non-numerical methods. For complex spatially distributed models, the parameter search space is usually very large, and none of direct search or gradient based methods works well. The non-numerical methods are gaining popularity in solving these types of large-scale optimization problems.

In current study, we build our optimization service on an improved genetic algorithm [8, 6]. Other optimization methods can also be included if needed in the future. The optimization service maintains the control parameters of the genetic algorithm together with other configurations, and provides access to the algorithm operations based on the standard SOAP protocols. The interface of the optimization service is illustrated in Fig. 1. Because there isn't any communication among computation servers and the overhead of the system is really low, nearly-linear speedup can be achieved, and running the job in the Grid environment is similar to running the job in a single cluster. Hence, the system is highly scalable. The detailed description of the improved genetic algorithm is beyond the scope of current paper, and is the subject of another paper currently under preparation.

4. APPLICATION

North China Plain (NCP) is one of the most important agriculture and industrial bases in China. Decades of rapid economic and population growth has brought a severe water crisis and many related environmental problems to this region. To achieve proper management of water resources and sustainable development, the point is to understand the dynamics of water cycle in this system. In current project, we use a simulation model (SWAP) [9] to study the water movement in an integrated soil-water-atmosphere-plant system, and apply GGPE-G to the modeling system to perform parameter estimation.

The one-dimensional SWAP model uses the Richard's equation (Equation 1) to model the physical process of water movement in unsaturated soil layers.

$$\frac{\partial \theta}{\partial t} = C(h)\frac{\partial h}{\partial t} = \frac{\partial \left[K(h)\left(\frac{\partial h}{\partial z}+1\right)\right]}{\partial z} - S_a(h) \qquad (1)$$

In our application, we adopt the analytical function proposed by Van Genuchten (Equation 2) [13], and Mualem's $K(\theta)$ function (Equation 3) [10] to represent the functional relationships between the water content θ, the pressure head h and the hydraulic conductivity K.

$$\theta = \theta_{res} + \frac{\theta_{sat} - \theta_{res}}{(1 + |\alpha h|^n)^m} \qquad (2)$$

$$K = K_{sat} S_e^{\lambda} \left[1 - \left(1 - S_e^{\frac{1}{m}}\right)^m\right]^2 \qquad (3)$$

where θ_{sat} is the saturated water content ($cm^3 cm^{-3}$), θ_{res} is the residual water content in the very dry range ($cm^3 cm^{-3}$), K_{sat} is the saturated conductivity (cmd^{-1}), λ is a shape parameter (-). α (cm^{-1}), n (-) and $m = 1 - \frac{1}{n}$ (-) are empirical shape factors. S_e is the relative saturation defined as:

$$S_e = \frac{\theta - \theta_{res}}{\theta_{sat} - \theta_{res}} \qquad (4)$$

These soil hydraulic parameters, θ_{sat}, θ_{res}, K_{sat}, λ, α and n, need to be specified for each distinct soil layer. In our simulations, there are 25 different soil layers. With 6 different parameters within each layer, we have in total $150(25 \times 6)$ unknown parameters need to be estimated. The conventional direct search or gradient based techniques will not be feasible choices for parameter estimation in this case, due to the strong nonlinearity of the Richard's Equation and the huge search space created by the large number of unknown parameters. It may takes the conventional methods up to 10 years to search for the optimal estimates in a system with single Intel Xeon 3.0GHz processor. Here, we attempt to tackle this program with Grid computing based on an improved generic algorithm.

Field data used for model calibration were collected at Luancheng Agro-Ecological Research Station (Chinese Academy of Sciences), Luancheng County, Hebei Province, from October 1998 through September 2000. The data include meteorological records, crop properties, and soil classifications that serve as the inputs to the model, and also include the soil moisture change over 2 years that are the targets of model calibration. Winter wheat is planted from October through June, and maize from June through September, according to local cropping practices. The soil moisture contents are measured by a neutron probe approximately every five days at nine to ten depth intervals between 0 and 180 cm.

We have set up GGPE-G in the Grid platform of Guangdong Key Laboratory of Computer Network, South China University of Technology. The SWAP model are wrapped as a computation service that runs on these servers. Computation service acts as an intermediary for the SWAP model by exposing an interface to users. The function of the computation service is to listen for SOAP calls and then call the SWAP model with appropriate input files. Figure 4 illustrates the role of the computation service in the system. In the context of running the SWAP model, the input files

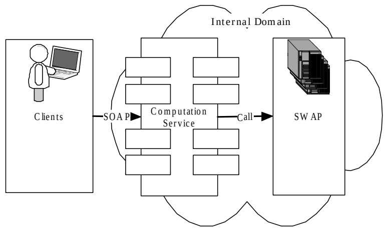

Figure 4: Computation service serve as an intermediary between invokers and SWAP model

include the parameters to be estimated, boundary conditions, meteorological records, irrigation schedule, and crop properties etc.. Rather than relying on clients to provide appropriate input files, we simply expose interfaces that allow automatic generation of input files with key data from clients. These input files are then forwarded to the SWAP model for further execution.

Users can login to the Grid platform and access to the services described in section 2.1. By providing a configuration file, users can submit jobs to the Grid through Grid Resource Allocation and Management (GRAM) component, which comprises a set of Web services to locate, submit, monitor, and cancel jobs on Grid computing resources. The internal invocation process and overall procedure can be found in figure 2 and section 2.3.

We have carried out the parameter estimation job as described in previous paragraph in 82 heterogeneous servers with total 208 CPUs. The processor architectures of these servers include i686, x86_64 (AMD Opteron), IA64 (Itanium 2) and Sun Ultra Sparc. The operating systems include Linux and Solaris. The entire parameter optimization process lasts 15 days. As described in section 3.4, the system can achieve nearly-linear speedup, so we can see that about

8.5 years is required to accomplish this job in single processor.

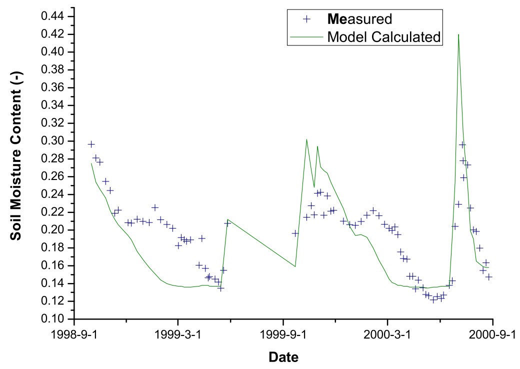

Figure 5: Comparison between measured and model calculated soil moisture content at the depth of 15cm at site 6

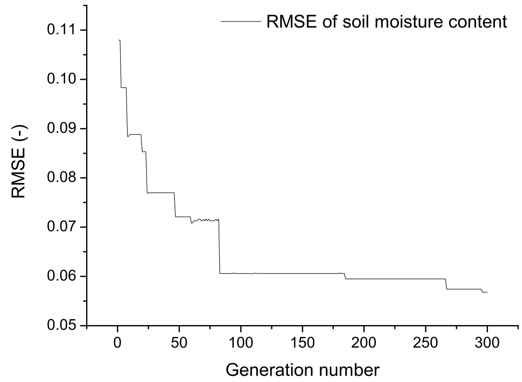

Figure 6: Convergence of soil moisture content's RMSE during parameter estimation process

Figure 5 gives a sample of the comparisons between the measured and model calculated soil moisture changes at the interval of 15cm below ground surface. Overall, the RMSE between measured and model calculated soil moisture content of top 180cm was 4.9cm, or 12.5% of the averaged total water content. Figure 6 illustrates the convergence of RMSEs of soil moisture content during the parameter estimation process.

5. CONCLUSION

In this paper we describe the architecture, implementation and application of a service-oriented parameter estimation system, GGPE-G. By coupling Grid technology with genetic algorithm, GGPE-G enables a new way to tackle large-scale parameter estimation problems, whose solution are normally impractical with the conventional approaches. With a service oriented architecture, GGPE-G can be easily deployed, efficiently monitored and effectively used. In a complex environmental modeling with a large number of unknown parameters, we demonstrate the utility of GGPE-G with encouraging results.

However, the scheduling of GGPE-G is currently driven by a simple greedy algorithm and the performance comparison comes from rough estimation. More advanced schedulers and detailed performance analysis will be our future work.

6. ACKNOWLEDGEMENTS

This research was jointly funded by Li Zheng's "Young Talents Award", the Innovation Knowledge Project of Chinese Academy of Sciences (project No. KZCX3-SW-428) and the ChinaGrid Project (project No. CG2003-GA002 and CG2003-GA005).

7. ADDITIONAL AUTHORS

Additional authors: Liqun Tang (College of Traffic and Communications, South China University of Technology, Guangzhou, 510641, China, email: lqtang@scut.edu.cn).

8. REFERENCES

[1] Geodise project. http://www.geodise.org/.
[2] Globus project. http://www.globus.org/.
[3] Resource Aware Visualisation Environment (RAVE) project. http://www.wesc.ac.uk/projectsite/rave/.
[4] Spitfire project. http://edg-wp2.web.cern.ch/edg-wp2/spitfire/, 2005.
[5] J. Doherty. *PEST: Model-Independent Parameter Estimation*. Watermark Numerical Computing, fifth edition, 2004.
[6] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, Reading, Mass., 1989.
[7] A. Harbaugh, E. Banta, M. Hill, and M. McDonald. Modflow-2000, the u.s. geological survey modular ground-water model – user guide to modularization concepts and the ground-water flow process. Open-File Report 00-92, U.S. Geological Survey, 2000.
[8] J. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, MI, 1975.
[9] J. Kroes and J. van Dam. *Reference Manual SWAP version 3.0.3*. Alterra, Green World Research, Wageningen, The Netherlands, 2003.
[10] Y. Mualem. A new model for predicting the hydraulic conductivity of unsaturated porous media. *Water Resources Research*, 12:513–522, 1976.
[11] E. P. Poeter and M. C. Hill. Documentation of UCODE: A computer code for universal inverse modeling. Water-Resources Investigations Reports 98-4080, U.S. Geological Survey, 1998.
[12] G. E. Pound, M. H. Eres, J. L. Wason, Z. Jiao, A. J. Keane, and S. J. Cox. A grid-enabled problem solving environment (pse) for design optimisation within matlab. In *IPDPS*, page 50. IEEE Computer Society, 2003.
[13] M. Van Genuchten. A closed form equation for predicting the hydraulic conductivity of unsaturated soils. *Soil Science Society American Journal*, 44:892–898, 1980.